CS640	**Computational Complexity**

Random Walks on Expanders

Instructor: Manindra Agrawal *Scribe: Ramprasad Saptharishi*

Last class we saw how linear algebra can be used; we saw that spectral expansion and vertex expansion are essentially equivalent.

To illustrate yet another example of how we can use linear algebra to prove theorems on expanders, let us prove the theorem we informally stated in the earlier lecture - amplifying success probability in an RP algorithm.

We stated that if we consider a small subset and took a random walk starting from a vertex in this subset, then the probability that we don't leave this set is exponentially small. We shall prove this formally this class.

1 Random Walks on Expanders Graphs

Theorem 1. *Let S be an arbitrary subset of vertices of a (n, d, λ) expander of size at most δn. Pick a vertex X_1 at random and start a walk of length k from that vertex. Let the walk be $X_1, X_2, \cdots, X_{k+1}$. Then*

$$\Pr[X_1, X_2, \cdots X_{k+1} \in S] \leq ((1-\lambda)\delta + \lambda)^k$$

Proof. We shall express all probabilities, etc in terms of matrix/vector products and analyze them. We first pick a vertex at random and therefore any vertex of the graph can be picked with probability $1/n$. Let us capture this by using a vector $|p_1\rangle = \frac{1}{n}\sum_{i=1}^{n}|i\rangle$ that has a value $1/n$ in every coordinate.

We want the entire walk in the set S and therefore we want the first vertex X_1 also to belong to the set S. In order to capture this, we need to rule out the cases when the vertex does not belong to S. This can be captured by the diagonal matrix, say B, that has a 1s exactly at all (i,i)-th positions when $i \in S$ and the rest of the entries 0. Multiplying any vector with this matrix would preserve just the coordinates corresponding to S and kill all other coordinates to 0. In Dirac's notation

$$B = \sum_{i \in S} |i\rangle \langle i|$$

Hence, if we want X_1 to belong to S, we need to sum up all the coordinates of $B\left|p_1\right\rangle$ which is just the L_1 norm of $B\left|p_1\right\rangle$. Then

$$\Pr[X_1 \in S] = \left|B\left|p_1\right\rangle\right|_1$$

What about the next vertex? Notice that if you start at some probability vector v which represents the current distribution over vertices, the new distribution after choosing a neighbour at random is just Av. And now we need to check if this new vertex is in S and therefore

$$\Pr[X_1, X_2 \in S] = \left|BAB\left|p_1\right\rangle\right|_1$$

And similarly, for a walk of k steps, the final probability would just be

$$\Pr[X_1, X_2, \cdots, X_{k+1} \in S] = \left|(BA)^k B\left|p_1\right\rangle\right|_1$$

It is always useful to convert it into L_2 norms since it is easier to analyze than compared to the L_1 norm. Let $\left|\hat{1}\right\rangle$ be the vector with every coordinate as 1. Then $\left|v\right|_1 = \left\langle\hat{1}\right| \left|v\right\rangle$ and therefore by applying Cauchy Schwarz, we get $\left|v\right|_1 = \left\langle\hat{1}\right| \left|v\right\rangle \leq \sqrt{n}\left\|v\right\|$.

Here is one possible approach: we have a notion of a norm of a vector. What about the norm of a matrix? Thinking of a matrix as a transformation taking one vector to another, can amount by which the matrix "stretches" the vector be used to define the norm of a matrix?

Definition 1. *For an $n \times n$ matrix C, the norm of C is said to be the least $\alpha \geq 0$ such that for every vector $\left|v\right\rangle$, $\left\|C\left|v\right\rangle\right\| \leq \alpha \left\|\left|v\right\rangle\right\|$.*

Thus we could use this definition to get an upper bound on $\left\|(BA)^k B\left|p_1\right\rangle\right\|$ by eliminating one matrix at a time. Let us try and see what the norm of B and A are.

The norm of B is 1 since if you give it a vector that has non-zero entries at exactly those coordinates corresponding to S, then B would just preserve the vector. Therefore, the norm of B is 1.

What about A? Unfortunately, the norm of A is 1 as well. Thinking of vectors expressed as linear combinations of the eigenbasis, notice that except the direction of $\left|v_1\right\rangle$, A shrinks every other direction since $\left|\lambda_i\right| \leq 1$. But since A preserves $\left|v_1\right\rangle$, the norm of A is 1 as well.

But then this would give us something as trivial as $Pr[X_1, X_2, \cdots, X_k \in S] \leq 1$. But the point is that we learn something important from this attempt:

- The matrix B kills all components of the vector in directions outside S.
- The matrix A shrinks all but the component along $\left|\hat{1}\right\rangle$ by at least λ.

So instead, let us look at the norm of the matrix BA. At this point we shall use a nice trick. Write the matrix $A = (1-\lambda)J + \lambda C$, where J the $n \times n$ matrix with every entry being $1/n$. One could express J as $\frac{1}{n}\left\langle\hat{1}\right|\left|\hat{1}\right\rangle$ in the dirac notation. We would be using the following two important properties of the norm.

Observation 2. $\|P+Q\| \leq \|P\| + \|Q\|$ *and* $\|PQ\| \leq \|P\|\,\|Q\|$

Proof. Suppose $\|P\| = \alpha$ and $\|Q\| = \beta$. Then:

$$\begin{aligned} \|(P+Q)\left|v\right\rangle\| &= \|P\left|v\right\rangle + Q\left|v\right\rangle\| \\ &\leq \|P\left|v\right\rangle\| + \|Q\left|v\right\rangle\| \\ &\leq \alpha\,\||v\rangle\| + \beta\,\||v\rangle\| \\ &\leq (\alpha+\beta)\,\||v\rangle\| \end{aligned}$$

The other property is pretty straight forward as well. □

Using this, we have $\|BA\| \leq (1-\lambda)\,\|BJ\| + \lambda\,\|BC\|$. Note that for any vector $|v\rangle$, $J\left|v\right\rangle$ will just have every coordinate as the average of all entries of v. Let us say the average of all entries of v was t. Then $BJv = B(t\left|\hat{1}\right\rangle)$. Now B would kill all but the coordinates that belong to S and therefore $\|B\left|v\right\rangle\| = t\sqrt{|S|}$. Therefore

$$\|BJ\left|v\right\rangle\| = t\sqrt{S} = t\sqrt{\delta n} = \frac{||v\rangle|_1}{n}\sqrt{\delta n} \leq \frac{\sqrt{n}\,\||v\rangle\|}{n}\sqrt{\delta n} = \sqrt{\delta}\,\||v\rangle\|$$

We need to get a bound on $\|BC\|$ and we would be done.

Claim 3. $\|C\| \leq 1$.

Proof. By our definition, $C = \frac{1}{\lambda}\left(A - (1-\lambda)J\right)$. Let $|v\rangle$ be any vector. Write $|v\rangle = |v'\rangle + |w\rangle$ where $|v'\rangle$ is parallel to $\left|\hat{1}\right\rangle$ and $|w\rangle$ is orthogonal to it.

$$C\left|v'\right\rangle = \frac{1}{\lambda}A\left|v'\right\rangle - \frac{1-\lambda}{\lambda}J\left|v'\right\rangle = \frac{1}{\lambda}\left|v'\right\rangle - \frac{1-\lambda}{\lambda}\left|v'\right\rangle = \left|v'\right\rangle$$

As for the orthogonal component, notice that $J\left|w\right\rangle = 0$ since w is orthogonal to $\left|\hat{1}\right\rangle$. Therefore,

$$C\left|w\right\rangle = \frac{1}{\lambda}\left(A\left|w\right\rangle - (1-\lambda)J\left|w\right\rangle\right) = \frac{1}{\lambda}A\left|w\right\rangle$$

Taking norms,

$$\|C\,|w\rangle\| = \frac{1}{\lambda}\,\|A\,|w\rangle\| \leq \||w\rangle\|$$

since $|w\rangle$ is orthogonal to $\left|\hat{1}\right\rangle$. Also note that the vector $C\,|w\rangle$ will be a vector in the orthogonal complement of $\left|\hat{1}\right\rangle$.

$$\begin{aligned}\|C\,|v\rangle\|^2 &= \left\|C\left|v'\right\rangle\right\|^2 + \|C\,|w\rangle\|^2 \\ &\leq \left\|\left|v'\right\rangle\right\|^2 + \||w\rangle\|^2\end{aligned}$$

And therefore, $\|C\| \leq 1$. It is infact equal to 1 since the second eigenvector $|v_2\rangle$ is the candidate. □

And therefore,

$$\begin{aligned}\|BA\| &\leq (1-\lambda)\,\|BJ\| + \lambda\,\|BC\| \\ &\leq (1-\lambda)\sqrt{\delta} + \lambda \\ \implies \left\|(BA)^k B\,|p_1\rangle\right\| &\leq \left((1-\lambda)\sqrt{\delta} + \lambda\right)^k \|B\,|p_1\rangle\| \\ &= \left((1-\lambda)\sqrt{\delta} + \lambda\right)^k \sqrt{\frac{|S|}{n^2}} \\ &= \left((1-\lambda)\sqrt{\delta} + \lambda\right)^k \sqrt{\frac{\delta}{n}} \\ \implies \left|(BA)^k B\,|p_1\rangle\right|_1 &\leq \sqrt{n}\cdot\left((1-\lambda)\sqrt{\delta} + \lambda\right)^k \sqrt{\frac{\delta}{n}} \\ &= \left((1-\lambda)\sqrt{\delta} + \lambda\right)^k \sqrt{\delta} \\ &\leq \left((1-\lambda)\delta + \lambda\right)^k\end{aligned}$$

And that proves the theorem. □

2 Amplifying Success Probability in Randomized Algorithms

We did make a passing remark on how expanders can be used to reduce the number of random bits used in the naive boosting technique to boost the success probability.

2.1 Expanders in RP algorithms

Let us look at any RP algorithm; probabilistic algorithm with error only on one side. When x was in the language, then your algorithm always answers yes on all random strings. The only case when you could be fooled is when x is not in the language but your random choice also says yes. The naive approach is to repeat this say k times and hope that you get at least one random choice that says no.

Suppose the algorithm had error probability of say δ initially and used m random bits. Then after k tries, the error probability as δ^k and the total number of random bits would be mk. We want to make this more efficient.

The idea is to interpret each random m-bit string as a vertex in an expander graph. Consider an $(2^m, d, \lambda)$ expander and each vertex encodes a random string. You pick a vertex at random in the graph. That corresponds to some string. Run that algorithm on this string. Then pick a random neighbour of this vertex and move to this vertex. Use the string corresponding to this vertex as your new random string and do the algorithm. Repeat this k times by making a walk of length k on the expander.

Let S be set of random strings on which your algorithm errs. Then since the error probability was δ to start with, $|S| = \delta n$ and we can apply the above theorem. Then we get the error down to exponential in k by a k-step random walk.

The total number of random bits is m to get the first vertex, and then $\log d$ random bits to pick a random neighbour everytime. Therefore, the total number of random bits is just $O(m+k \log d)$ and d is usually a constant and therefore $O(m + k)$ which is way better than $O(mk)$.

2.2 Expanders in BPP algorithm

In order to reduce the number of random bits in a BPP algorithm we need a stronger theorem on random walks to apply for BPP. We won't be doing the proof of this; it is far more complex than the one we proved above.

Theorem 4. *Let S be a subset of vertices of an (n, d, λ) expander graph such that $|S| \leq \delta n$. Let $X_1, X_2, \cdots, X_{k+1}$ be a random walk on this expander graph. Then*

$$\Pr\left[\left|\frac{\#\{i \ : \ X_i \in S\}}{k} - \delta\right| > \epsilon\right] \leq 2e^{-\frac{(1-\lambda)\epsilon^2\delta}{60}}$$

This essentially means that if you start with say an error probability of $1/4$ initially, then essentially after a k step random walk, the number of

times you revisit the set of bad vertices is very close to δk. Therefore, if you started with $1/4$, your final fractino will also be very close to $1/4$ and therefore, the majority vote will will give you the right answer with prob $2^{-O(k)}$.

Thus, with $O(m+k)$ random bits, you can amplify the success probability in a BPP algorithm as well.